



A Real-Time Machine Learning-Based Road Safety Monitoring and Assessment System

Tulsi Pawan Fowdur¹ · Mohammed Fayez Hawseaa¹

Received: 1 August 2023 / Revised: 14 March 2024 / Accepted: 20 March 2024 / Published online: 28 March 2024
© The Author(s), under exclusive licence to Intelligent Transportation Systems Japan 2024

Abstract

In this paper a system to monitor road conditions, detect unsafe driving behaviours and determine the influence of rainfall on traffic safety in real time using different machine learning algorithms, has been proposed. The system developed consists of a mobile application that captures car movement using its in-built accelerometer and gyroscope sensors and a server that monitors weather conditions at 16 key locations in Mauritius using the OpenWeather API. Road conditions, pothole, speed bumps as well as driving events were analysed using the K-Nearest Neighbour (KNN) and Multi-Layer Perceptron (MLP) algorithms. Moreover, a mathematical model, which incorporates the predicted rainfall in the estimation of braking distance and recommended speed, has been proposed. An average accuracy of 80.9% was obtained for pothole detection, 70% for speed bumps and 85.5% for unsafe driving behaviours detection. The proposed model with rainfall data predicted the braking distance and recommended speed with a Mean Absolute Percentage Error (MAPE) of 14.7% and 0.735% respectively.

Keywords Traffic safety · Braking distance · Rainfall forecasting · Potholes · Unsafe driving behaviours · Speed bumps · Machine learning · Real-time

1 Introduction

Traffic safety holds an important place in our lives as it ensures the well-being and protection of individuals on the road. The impacts of road accidents should not be overlooked as in many cases, they may lead to financial losses, injuries or even loss of lives. In Mauritius, the number of registered road accidents for the first semester of 2022 was 37% higher than in the first semester of the previous year [1]. According to the World Health Organization (WHO), about 1.3 million people lose their life due to road accidents each year and between 20 and 50 million additional individuals suffer from non-fatal injuries [2]. This indicates that measures need to be taken to remediate the situation. In Sweden, a new road layout called “2–1 road” was introduced. It consists of two wide shoulders with a two-way lane in between which allows overtaking in a safe way [3].

Authorities in Australia have installed wire rope safety barriers which makes car crashed more forgiving [4]. However, these projects tend to be costly and time-consuming. A better approach would be to integrate a system for monitoring real-time traffic safety by considering factors such as road surface conditions, driving behaviours and the influence of weather.

Several studies have been conducted to implement such a system by making use of machine learning techniques. In [5], a system for detecting potholes using images from the phone camera is proposed. The system uses CNN to process the images and a modified VGG16 backbone to balance computation speed and accuracy. Similar work using images for detecting potholes is proposed in [6]. The proposed scheme made use of thermal imaging to extend pothole detection at night when thermal vision is not available. Studies carried out in [7, 8] proposed pothole detection schemes using accelerometers and gyroscopes from phone in-built sensors. These types of systems proved to be highly accurate. In [9], speed bumps were detected using sensor data collected from a raspberry pi and making use of a logistic model created from statistical features. Another system for detecting speed bumps using LSTM is proposed in [10]. To identify unsafe driving behaviours the work proposed in [11] made use of sensors present in

✉ Tulsi Pawan Fowdur
p.fowdur@uom.ac.mu

Mohammed Fayez Hawseaa
mohammed.hawseaa@uemail.uom.ac.mu

¹ Department of Electrical and Electronic Engineering,
University of Mauritius, Réduit, Mauritius

the car. A list of descriptive features was identified to characterize the driver's behaviour. Feed Forward neural Networks and SVM were then used to classify the descriptive features. In [12], a classification framework which uses machine learning for detecting unsafe driving behaviours was developed. 2000 truck drivers were surveyed and the algorithms utilized were CART, RT AdaBoost and GBDT. The system was able to identify 9 types of unsafe driving behaviours. Similar works are carried out in [13–17].

In this paper, a real-time system consisting of a mobile application and a local server for monitoring road conditions and unsafe driving behaviours as well as estimating braking distance and recommending a safe driving speed during rainfall is presented. The mobile application is developed using android studio and is used to monitor sensor and location data which is sent to the local server to be saved in a MySQL database. Weather conditions at 16 locations are monitored by the server using the OpenWeather API. The mobile application displays a map containing the user's device location. Weather predictions based on the current location of the device can be requested. The server is a desktop application created using NetBeans and is used to perform all the computations. Based on the sensor data, potholes, speed bumps and unsafe driving behaviours are detected using KNN and MLP algorithms and the results are sent to the mobile application. MLP and KNN have proved to give good results for pothole detection in [8, 18], for speed bumps in [19], and unsafe driving behaviours in [11, 20]. Hence, they have been employed in this paper. The braking distance and recommended speed are estimated using the predicted rainfall intensity and speed of the vehicle. Rainfall is predicted using MLR. At the end of the driver's journey safety ratings based on the performance of the driver and the quality of the road are calculated and sent to the mobile application.

The main research questions addressed in this paper are as follows:

- Q1. How to implement a system for detecting potholes, speed bumps and unsafe driving behaviours using machine learning in real-time using in-built smartphone sensors?
- Q2. What is the performance of KNN and MLP in terms of accuracy for the mentioned system?
- Q3. What is the influence of weather conditions such as rainfall on traffic safety?
- Q4. How can the braking distance and recommended speed be estimated based on the intensity of rainfall and the speed of a vehicle?
- Q5. How to give safety ratings based on the road quality and performance of a driver?

The main novelties of this work are as follows:

- A system which combines detection of potholes, speed bumps and unsafe driving behaviours in a single application is proposed.
- Previous works have investigated the effects of road conditions and weather on traffic safety but not jointly.
- A mathematical model for estimating braking distance from the speed of a vehicle and rainfall intensities is proposed. Previous works have investigated the variation of skid resistance with rainfall intensity and how braking distance is calculated using skid resistance. However, no work has been carried out to combine these two concepts.
- A mathematical model for recommending a safe driving speed during rainy weather is proposed.
- A new method for assigning ratings to a driver and the road based on the driver's performance and road quality is presented.

This paper is organized as follows. Section 2 provides a background review of previous works which have implemented systems for monitoring road conditions, unsafe driving behaviours and the influence of weather on traffic safety. Section 3 describes the implementation of the proposed system and outlines the processes involved. Section 4 evaluates the proposed system by performing tests on the algorithms used and interprets the results obtained. The expected capabilities of the system are also illustrated. Section 5 concludes the paper and provides possible future improvements to this work.

2 Related Works

In this section a review of pothole detection, speed bump detection and unsafe driving behaviours detection schemes using machine learning is presented. A review of schemes analysing the impact of rainfall on traffic is also given.

Most methods for detecting potholes using mobile devices rely on images captured from the phone camera. In [5], Khaled R. Ahmed presented a system for detecting potholes in real time which balances speed and accuracy. The system is automated and makes use of an efficient deep-learning convolutional neural network (CNN) for image processing. To achieve high accuracy and improve the computation speed, a VGG16 network was modified by eliminating some of the convolutional layers and making use of different dilation rates. The VGG16 backbone provided faster R-CNN when compared with other backbones such as MobileNetV2 and InceptionV3. The experimental results showed that this system provides an accuracy of 88%. In [6], a technique for detecting potholes using thermal imaging was proposed with the aim of finding the feasibility of such

a system. This approach extends existing pothole detection schemes to be able to operate at night when night vision is not provided. The system makes use of CNN and to train the model, images of potholes under different conditions and weather were collected. To increase the size of the training set, data augmentation techniques were applied. The experimental results showed that the system was able to classify images with an accuracy of 97.08%. In [7], vibration sensors and global positioning system receivers found

in a smartphone were used to detect potholes automatically. Several machine learning algorithms were analysed among which the Random Forest method gave the best results with a precision of 85%. It was also found that features extracted from frequency and time domains of the data collected performed better than other features used for pothole identification. The system was validated using datasets created from different types of roads and tested to see if it can be applied universally. Similarly, in [8] a pothole detection system using accelerometer and gyroscope sensors found in modern smartphones is proposed. The system relies on vibrations created by potholes which can be easily measured on the axis reading. A neural network was trained from the sensor data to distinguish potholes from non-potholes. Figure 1 illustrates the architecture used.

The results showed that the classification model had an accuracy of 94.78% which is suitable for creating an accurate and sensitive supervised model for pothole detection.

In [19], Johny Marques et al. proposed a method to identify and mark anomalies on the road namely speed bumps by utilizing a GoPro for image capture and several machine learning algorithms. Data was collected for different types and shapes of speed bumps and three machine learning classification models were selected namely Naive Bayes, Multi-Layer Perceptron, and Random Forest. The flowchart in Fig. 2 shows the methodology used.

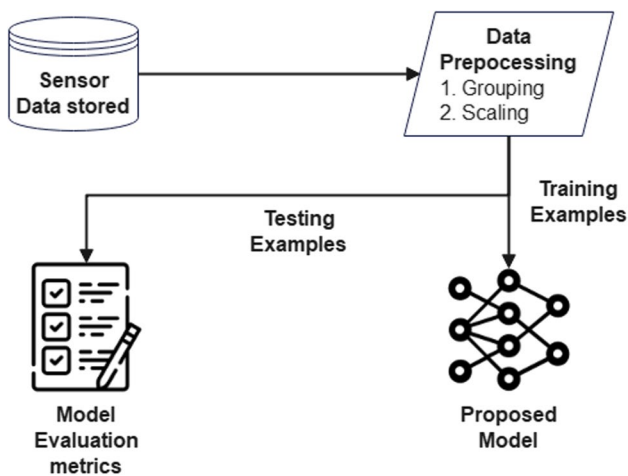
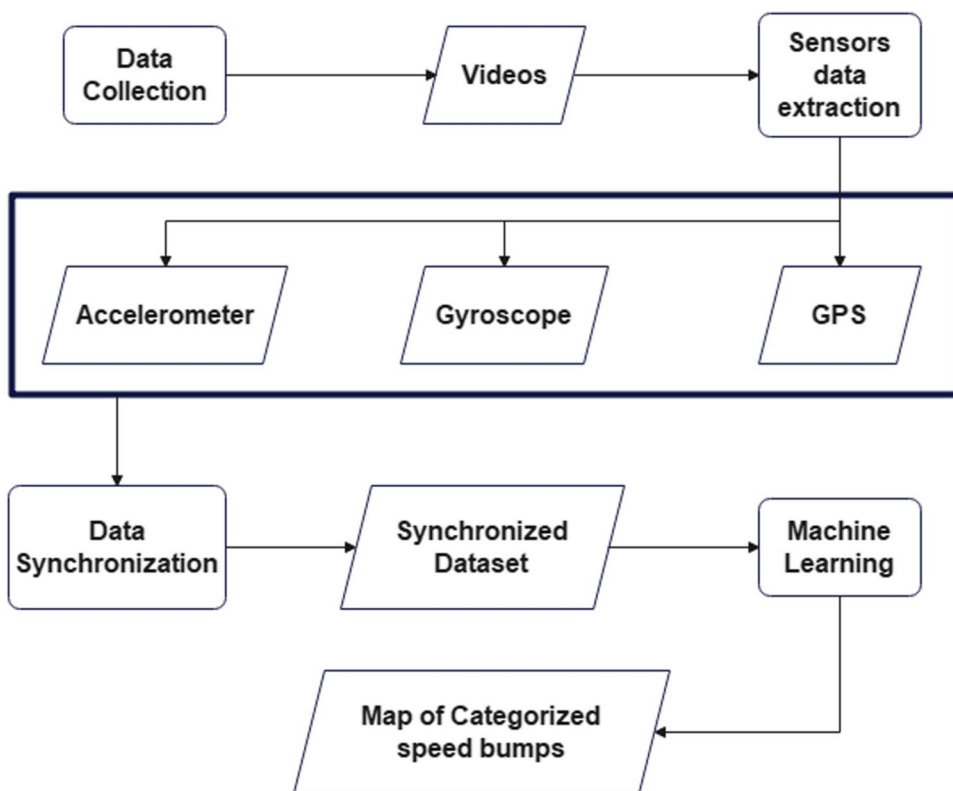


Fig. 1 Architecture for pothole detection system

Fig. 2 Flowchart for speed bump detection



The accuracy obtained for the three algorithms was above 96% and the system was able to generate precise maps of vertical road irregularities quickly with a fast update rate. In [9], speed bumps were detected by first collecting data from sensors such as accelerometer, gyroscope and GPS mounted on a car. The sensors were connected to a Raspberry Pi where statistical features which characterize the data are extracted. A machine learning approach was then used to find a logistic model that can detect speed bumps accurately. Results showed that the system could detect speed bumps with an accuracy of 97.14%. In [10], Vibrations caused by the vehicle passing over speed bumps are monitored. The 3-axis accelerometer data was collected and processed by a classification model. The algorithm used was LSTM as it has the capability of processing data over time while accelerometer data was continuously captured. The results showed an accuracy of 98% with minimal false positives cases.

In the study carried out in [11], a method for recognizing safe and unsafe driving behaviours using sensors present in the vehicle is proposed. A list of descriptive features to characterize the driver behaviour was created based on the following parameters:

- Engine speed
- Vehicle speed
- Engine load
- Steering wheel angle
- Throttle position
- Brake pedal pressure

Two classification algorithms namely Feed-Forward neural Networks Support Vector Machines (SVM) were selected for identifying the descriptive features. The classification models were able to classify the data with a mean accuracy above 90% which shows the capability of the system to identify different driving styles. In [12], a classification framework was developed to identify unsafe driving behaviours of truck drivers with the aim of reducing truck crashes. A survey was carried out among 2000 truck drivers to create the framework using machine learning. The machine learning algorithms considered were CART, RT AdaBoost and GBDT. The models consisted of six first-level input dimensions and 51 s-level input indicators related to proactive and objective factors. Nine types of unsafe driving behaviours were identified. Figure 3 shows the framework used.

The results showed that the accuracy of the models varies from 64 to 95%.

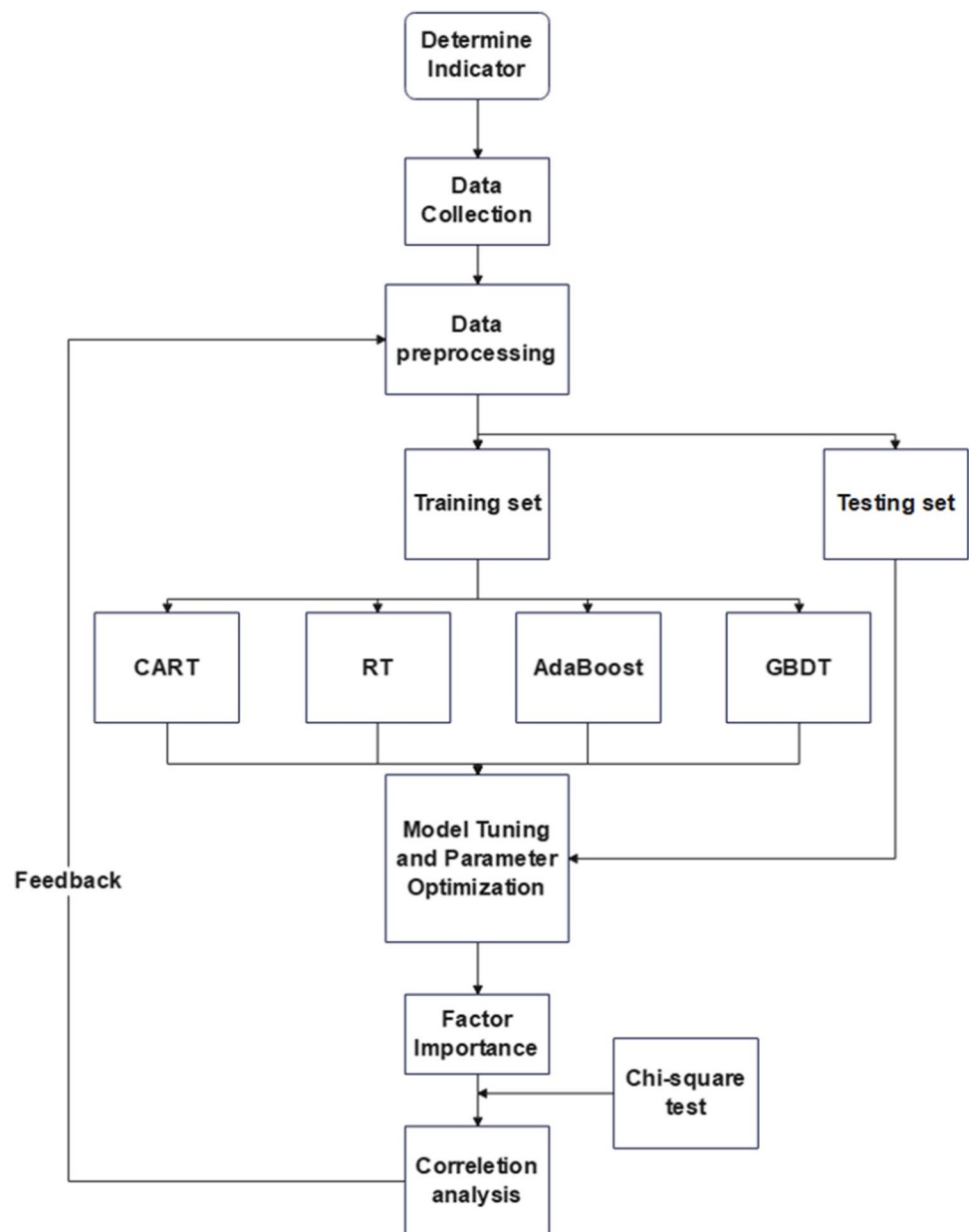
Prediction of rainfall has been carried out in several works using data mining and machine learning techniques. In [13] daily rainfall intensity was predicted using algorithms such as Random Forest, Multivariate Linear

Regression (MLR) and Extreme Gradient Boost. The atmospheric features which influence rainfall were identified using the Pearson correlation technique. These features were then used as input to the regression models. The performance of the algorithms was measured using the Root Mean Square Error and the Mean Absolute Error. The results showed that Extreme Gradient Boost performed better than the other algorithms. In [14], a prediction model using Long Short-Term Memory (LSTM) was proposed. Meteorological data was first collected and pre-processed. The pre-processing includes removing missing values, eliminating empty fields and normalizing the data. The deep-learning model was then trained from the data and tests were carried out to evaluate the performance of the algorithm. From the results obtained, it was concluded that LSTM performed better than machine algorithms such as MLP, KNN and SVM. In [15], Fowdur et al. presented a real-time weather forecasting system with collaborative regression. Weather data was collected using the OpenWeather API from a smartphone and desktop device for 4 different regions. Collaborative regression was then applied using five algorithms namely Multi Polynomial Regression (MPR), Multi Linear Regression (MLR), Multi-Layer Perceptron (MLP), K-Nearest Neighbours (KNN) and Convolutional Neural Network (CNN). The performance of each algorithm with and without collaborative regression was determined. The results showed that collaborative regression provides a MAPE which is 5% lower than non-collaborative methods. It was also observed that MLR gave better results than the other algorithms.

Several previous works have been carried out to determine the effect of weather conditions on traffic safety. In [16], the influence of changes in friction coefficient attributed to weather conditions is analysed. The dynamic motion of three types of vehicles namely bus, sedan and truck were investigated under different values of friction coefficients using an Adams/Car Simulator. The results showed that values of friction coefficients above 0.6 had no significant effect on braking distance and these values were attributed to dry weather conditions. As expected, values of friction coefficients 0.5, 0.4, 0.28, and 0.18 were attributed to wet, rainy, snowy and icy conditions respectively since these values had a consequent effect on braking distance.

In [17], skid resistance at different rainfall intensities was evaluated based on various pavement surface conditions, tyre thread design and the operating conditions of the tyre. Two types of roads namely Porous Asphalt (PA) and Dense Asphalt Concrete (DAC) were considered. The following tests were then carried out:

- Evaluate the effect of rainfall intensity on wet skid resistance
- How skid resistance varies for patterned tyres and smooth tyre

Fig. 3 Framework for unsafe driving behaviours detection

- The effect of various pavement cross slopes on wet skid resistance
- The effect of tyre-related characteristics such as inflation pressure, slip ratio and speed on skid resistance.

After the above was quantified, a reliable tool for evaluating skid resistance during rainfall was developed. The tool was then incorporated into pavement management systems so as to monitor highway traffic safety more accurately. Similarly, in [21] the variation of pavement friction during snowstorms in urban areas was investigated to determine its influence on traffic safety. Using weather

data collected hourly and road surface conditions information, negative binomial safety performance functions were created. It was found using statistics that the relationship between pavement friction level and traffic safety was considerable. Collisions occurred more frequently when the pavement friction was below 0.35 and less frequently when the pavement friction was above 0.6. The increase in collisions during snowstorms was attributed to the accumulation of snow and ice which degrades the road quality.

It is noted that previous works did not consider the integration of rainfall prediction in the determination of braking distance and optimal driving speed in real-time conditions.

3 System Model and Algorithms for Traffic Alerts

In this section, the complete system model which consists of a local server and mobile application is described. The algorithms used to perform pothole, speed bump and unsafe driving behaviours detection will be elaborated. The method for estimating braking distance and recommended speed during rainy weather will also be discussed. Finally, the calculations of road and driver rating for the driver’s journey will be discussed.

3.1 Complete System Model

Figure 4 shows a detailed diagram illustrating the components present in the system model and the interaction between each component.

The three main components present in the system are the local server, a mobile application and a local database. The local server performs weather predictions for the location of the user as detected by the smartphone and sends it to the mobile application when requested. It also performs classification for potholes, speed bumps and unsafe driving behaviours using the sensor data collected from the mobile

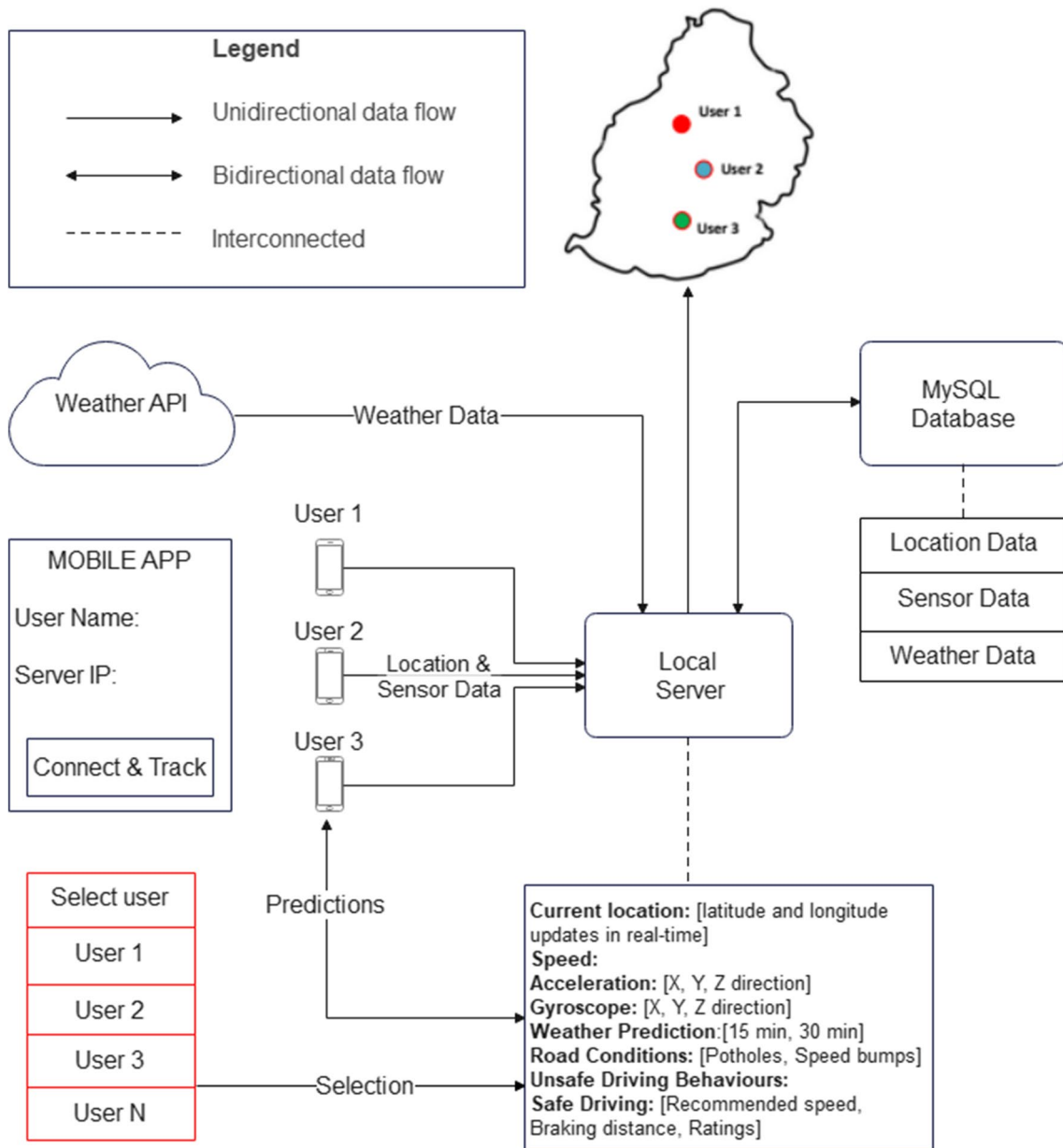


Fig. 4 Complete system model

application. A map is displayed for monitoring the geolocation of each connected device. The database is used to store weather data which is obtained from the OpenWeather API and sensor and location data sent by the smartphone. The mobile application receives the classification and weather forecasting results from the local server. A map is displayed so that the driver's position is indicated on a map. At the end of the driver's journey, ratings on the safety level of the journey are displayed on the mobile application.

3.1.1 Local Server

The local server is a desktop application created using the Java programming language. It accepts incoming connections from mobile client devices using a Server Socket connection. The functions of the local server are as follows:

- Accept incoming sensor and location data of the client devices and store them in a local SQL database.
- Monitor the current weather conditions at 16 different regions and store the data in a local SQL database. It is to be noted that 16 locations have been chosen as it is the maximum number of locations covered by the Weather API without exceeding the number of request limits per

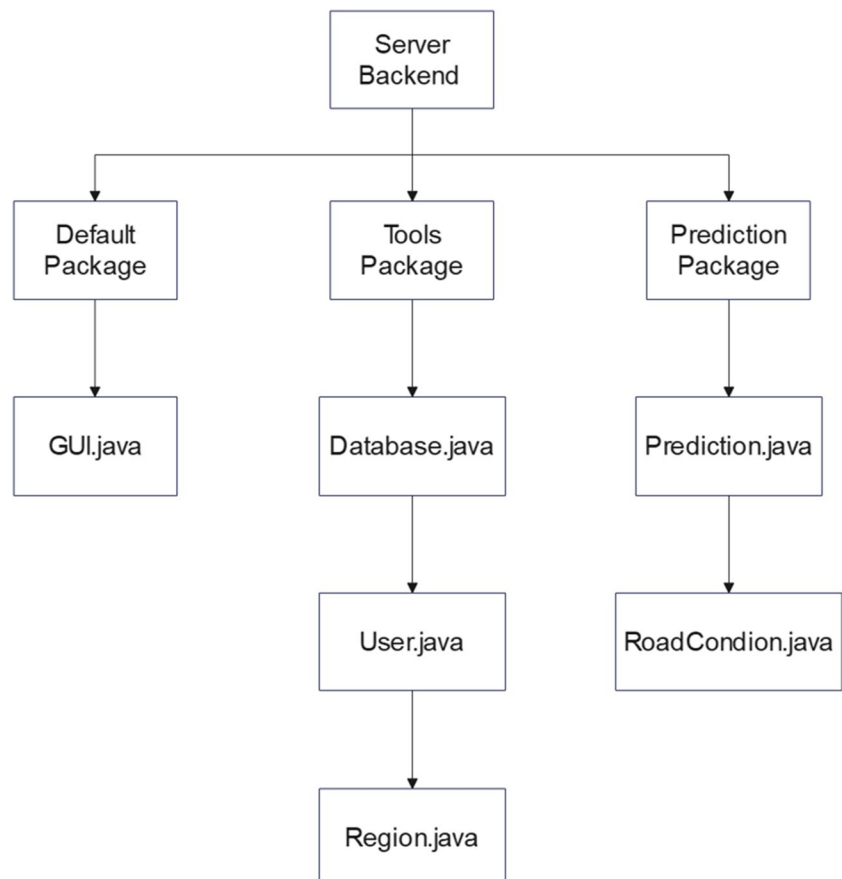
day and hence the maximum coverage it can provide for the northern part of Mauritius.

- Display a live map with markers indicating the geographic location of each client device.
- Perform real-time weather forecasting and relay the results to the client devices
- Determine the presence of potholes, and speed bumps and detect unsafe driving behaviours based on the sensor data before returning the results to the client devices.
- Estimate a safe driving speed and braking distance during rainfall and return the results to the client devices.
- Determine safety ratings of the driver's journey.
- Download requested weather, location or sensor data from the local SQL database.

The program structure of the local server consists of three distinct packages with a collection of Java classes. Figure 5 shows how the packages are organized with their corresponding classes.

The Default package contains the GUI.java class. The class is responsible for rendering all the components on the screen and has a container for displaying a map. It defines the logic when interacting with the GUI interface. It also handles all operations between clients and the server.

Fig. 5 Local server program structure



The Tools package contains three classes namely Database.java, User.java and Region.java. The Database.java class is used for writing and reading data from the local database. The User.java class is used to create an object which keeps track of all information pertaining to a user. The Region.java class initialized the 16 regions considered in Mauritius and holds their geographic location. Details about the selected locations are shown in Table 1. The weather conditions monitored were the cloudiness, temperature, humidity, pressure, wind speed, wind direction and amount of rainfall. An interval of 1 min was used between the recorded samples.

The prediction package consists of the Weather.java class and the RoadCondition.java class. Weather.java class is used to obtain weather predictions using MLR and collaborative regression. RoadCondition.java is used to detect potholes, speed bumps and unsafe driving behaviour using KNN and MLP classification algorithms. The class also estimates the braking distance and recommended speed based on the intensity of rainfall and the speed of the vehicle.

The weather prediction is implemented using MLR and collaborative regression. Pothole, speed bump and driving event detection are implemented using KNN and MLP algorithms. As for the braking distance and recommended speed, a mathematical model based on the adherence of the vehicle's tires to the road is used. The safety ratings give a subjective description of the road quality and the behaviour of the driver. The algorithms mentioned above are discussed in more detail in the upcoming sections. Figure 6 shows the layout of the desktop server application.

Figures 7 and 8 show the components present in the desktop application.

Table 1 Details of selected locations

Location	Latitude	Longitude
Goodlands	-20.035	57.643
Cap Malheureux	-19.984	57.641
Triolet	-20.055	57.545
Terre Rouge	-20.126	57.524
Riviere Du Rempart	-20.103	57.685
Grand Bay	-20.018	57.580
Plaine Des Papayes	-20.065	57.572
Port Louis	-20.162	57.499
Roches Noire	-20.111	57.712
Petit Raffray	-20.013	57.612
Laventure	-20.146	57.677
Long Mountain	-20.143	57.562
Pointe Aux Piments	-20.063	57.513
Grand Gaube	-20.006	57.661
Piton	-20.090	57.630
Pamplemousses	-20.104	57.570

3.1.2 Mobile Application

It is the end-user Android application that connects to the local server. The tasks performed by the application are as follows:

- Send real-time location and sensor information to the server
- Display the geographic position of the user on a map
- Request weather predictions
- Receive from the server a recommended driving speed and estimated braking distance based on weather and sensor data.
- Receive results of pothole and speed bump detection from the server.
- Get notified of unsafe driving behaviours from the server.
- Receive driver and road safety ratings of the journey.

The program structure of the mobile application consists of three Java classes. Figure 9 shows the Java classes present in the program structure.

The Connection.java class defines the actions performed in the connection Activity. The user enters a username and the IP address of the local server. The monitoring activity starts using an intent. A socket connection is then established between the mobile client device and the local server.

The Monitoring.java class is used to define all the functionalities of the monitoring activity. In this activity, the sensor data which includes the accelerometer and gyroscope values in the x, y, and z directions are continuously sent to the server via the established socket connection. The location data of the user which includes the speed, latitude, longitude, and bearing are also sent to the server via the same socket. The user can request the current weather conditions or weather predictions for the next 15 or 30 min by sending a request to the local server using the refresh button. The activity also contains text views for displaying the current speed, recommended speed during rainfall, and estimated braking distance. Furthermore, the presence of potholes, speed bumps, and unsafe driving behaviours are displayed to the user. The user can request ratings of the journey using the rating button.

The Map.java class defines the actions performed in the map activity. It shows the location of the user on a Google map as a blue dot using the current latitude and longitude values. The LocationListener interface is used to obtain location updates either through the network provider or the GPS provider. The location updates are then used to dynamically change the position of the user on the map as the user moves. The map contains other functionalities such as zooming, panning, and tilting.

Figures 10 and 11 show the layout of the Connection activity and Monitoring activity respectively.



Fig. 6 Layout of desktop server application

3.1.3 SQL Database

To store all the collected data, the SQL database provisioned by WampServer was used. It comes with a user-friendly environment called PhpMyAdmin which makes managing databases easy. The SQL database was chosen as it provides a tabular structure for storing data.

3.2 Data Collection Process

To be able to make predictions, machine learning algorithms need to have access to training data. For this project, data is obtained from three different sources. Firstly, weather data is obtained using the openWeather API. The second source of data is from the inbuilt sensors present

Fig. 7 Sub Layout 1 of desktop server application

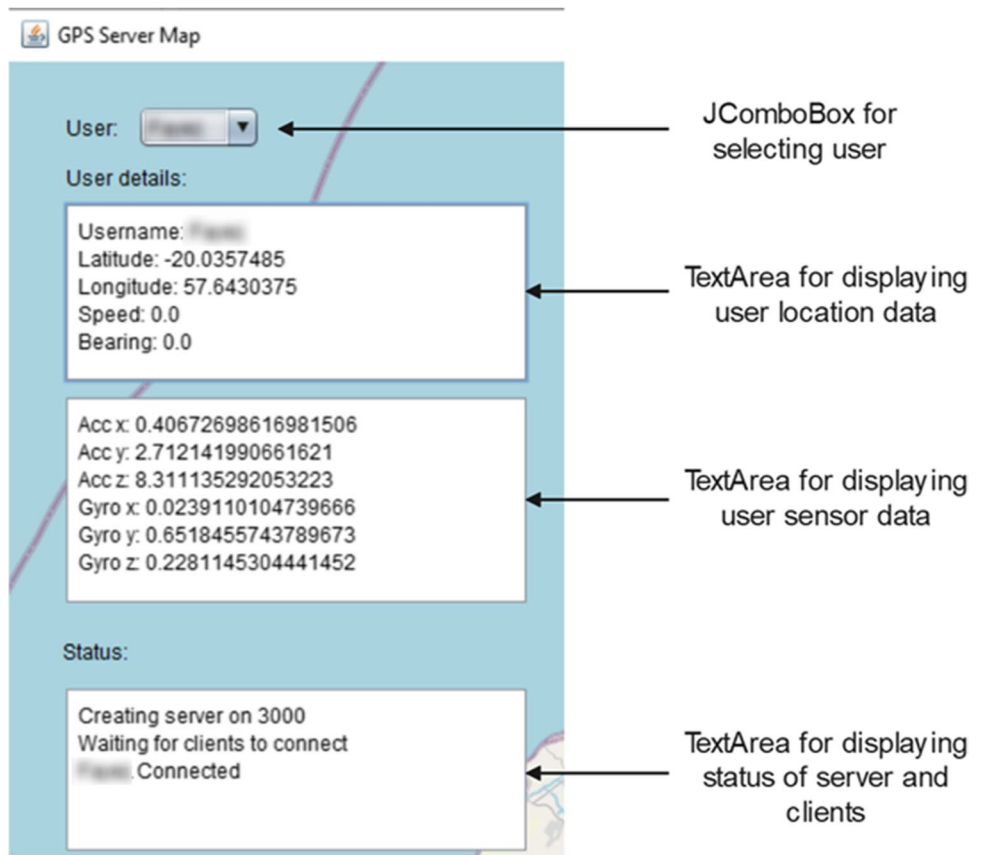


Fig. 8 Sub Layout 2 of desktop server application

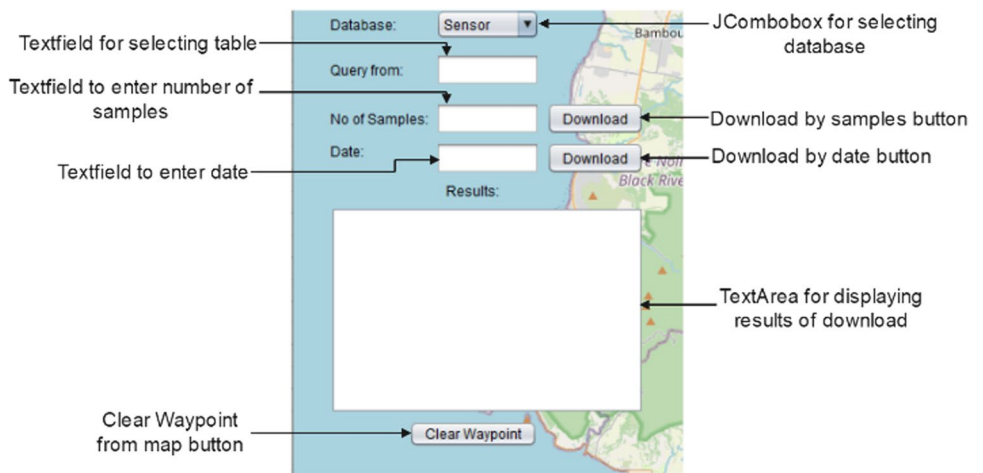
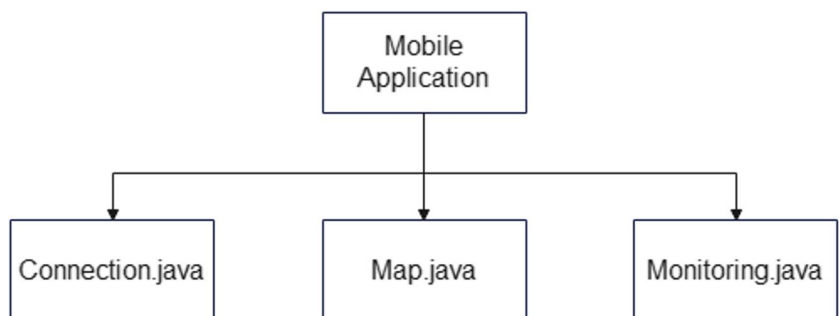


Fig. 9 Program structure of mobile application



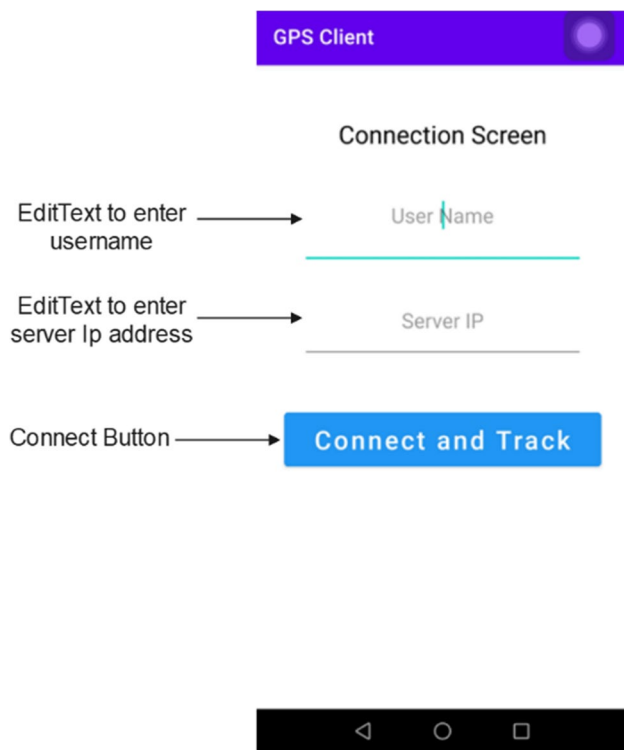


Fig. 10 Connection activity

in smartphones. Sensors that are used include the accelerometer and the gyroscope. To train some of the machine learning algorithms, pre-existing datasets were used. These datasets can be obtained from open-source websites such as Google Dataset Search, Kaggle, Hugging Face, and so on.

3.3 Weather Data

The weather conditions in the regions considered are obtained by calling the openWeather API every minute and storing it in the database. This process is executed on a separate thread for each region. An HTTP GET request with the necessary parameters in the URL is sent to the API endpoint. The parameters include the city name, API key, response type, and measurement units. In this case, the JSON format and metric units were used. The JSOUP library provides a convenient way for handling HTTP GET requests and extracting data from the response. The response is returned in a Document Object from which the data in JSON format is extracted. The weather conditions which are monitored are shown in Table 2.

Figure 12 shows a flowchart illustrating the weather data collection process.

3.3.1 Sensor Data

The Android application captures sensor data at a frequency of 0.2 s and transmits it to a local server for storage in a database. The process begins by creating a sensor manager, from which the Sensor objects for the accelerometer and gyroscope are obtained. These sensors are registered with a normal delay. In the `onSensorChanged()` method, the acceleration and gyroscope values are extracted. The sensor values are encapsulated in a JSON object which is subsequently sent to the local server via a socket connection. On the server side, the sensor values and the timestamp it was received are uploaded to the database. Figure 13 shows a flowchart of the sensor data collection process.

3.3.2 Existing Datasets

The dataset used for pothole detection was obtained from the Kaggle website [22]. It is a collection of several road trips which were carried out in the USA. Each road trip has a CSV file with the actual sensor values recorded. It also contains the timestamps that potholes were detected in a separate CSV file. A pre-processing was performed to combine the CSV files into a single file. The fields present in the combined CSV file are the acceleration and gyroscope values in the x, y, and z directions and the class which is either 0 or 1 meaning a pothole is present or not. The number of samples in the dataset is 9860. A section of the dataset is shown in Table 3.

The dataset used for determining unsafe driving behaviours was also obtained from the Kaggle website [23]. It models behaviours such as sudden acceleration, braking, left turns, and right turns. The dataset was collected for three drivers at the ages of 27, 28, and 37. The sampling rate was two samples per second and fields that are present in the dataset are mean, standard deviation, minimum, maximum, and current values of acceleration and gyroscope giving a total of 30 parameters. It was recommended to use a window size of 14 s. The number of samples is 2301. A section of the dataset is shown in Table 4.

As for the speed bumps, there were no datasets available online. So, a dataset was created by performing drive tests on the roads of Mauritius itself. The drive tests made use of an application for collecting sensor values when the vehicle went over the speed bumps. The sensor values were saved in a CSV file. The class is either 0 or 1 meaning a speed

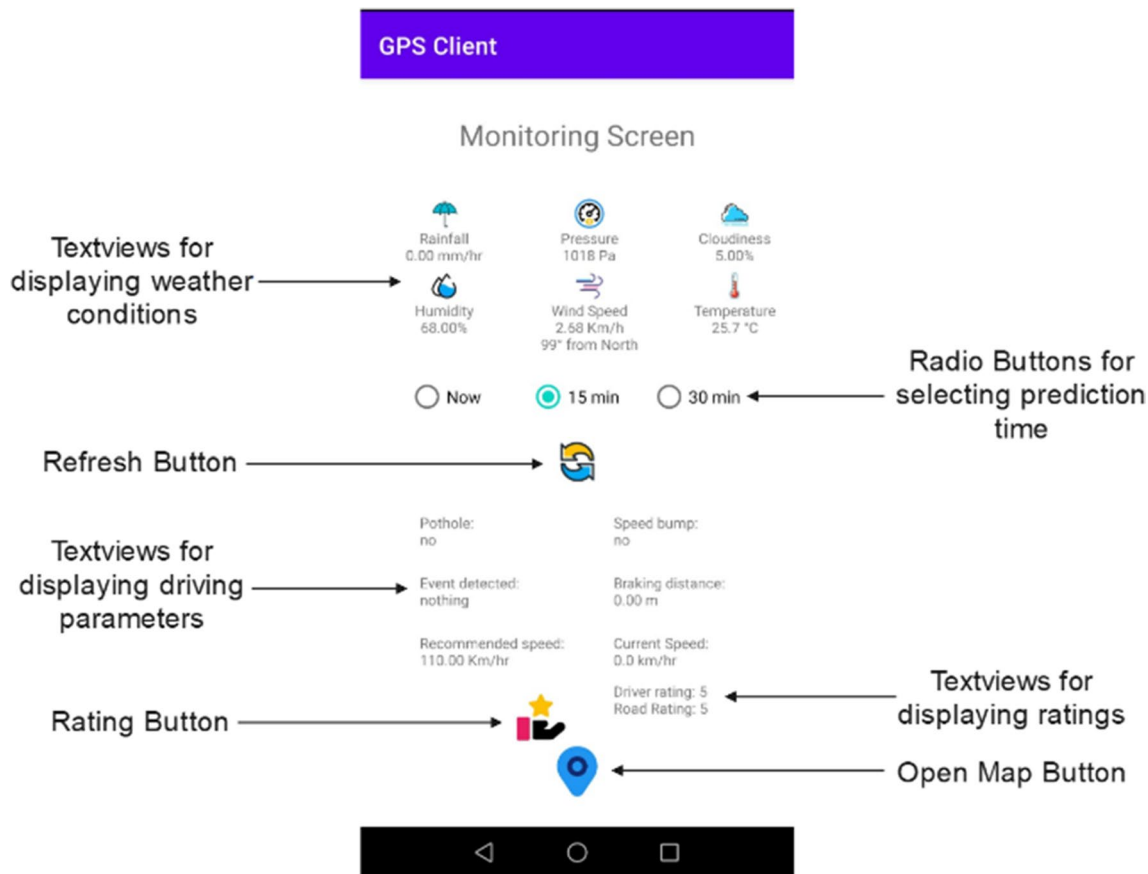


Fig. 11 Monitoring activity

Table 2 Weather conditions

Weather condition	Unit
Cloudiness	%
Temperature	Degree Celsius
Humidity	%
Rainfall	mm/hr
Pressure	Pa
Wind Speed	m/s
Wind Direction	Degree

bump is present or not. The number of samples is 660. A section of the speed bump dataset is shown in Table 5.

3.4 Machine Learning Algorithms for Traffic Safety

3.4.1 KNN Algorithm for Pothole, Speed Bump and Unsafe Driving Behaviours Detection

The K-Nearest Neighbours algorithm is a type of supervised machine learning algorithm that performs prediction or classification based on the closeness of data points. The

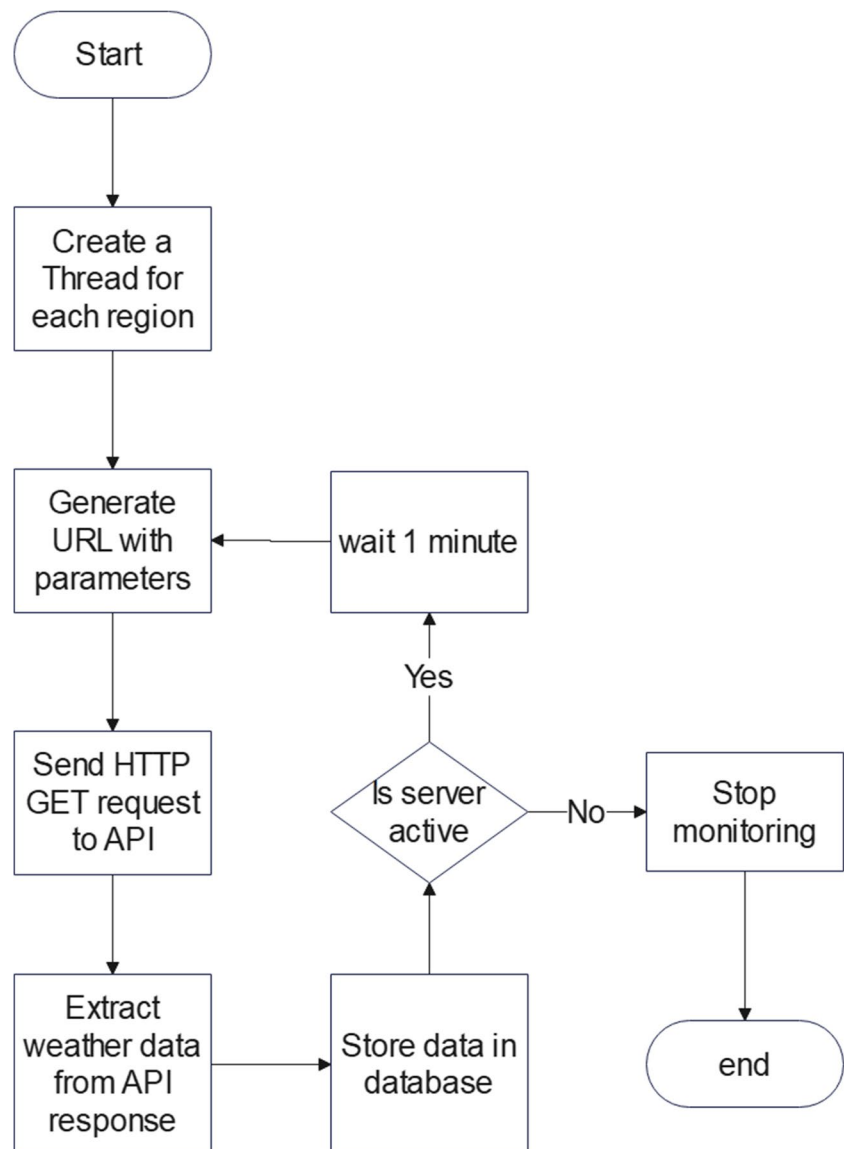
distance between the test instance and all the data points is calculated. Some of the distance measures are the Euclidean distance, Manhattan distance, and Minkowski distance. The K-nearest data points to the test data point is selected. The mode is chosen as the output when performing classification while the mean is calculated when performing regression [24, 25]. In this paper, the Euclidean distance was selected to implement the KNN algorithm. The expression for calculating Euclidean distance is shown in Eq. (1).

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2} \tag{1}$$

where,

- n is the number of features or dimensions
- x_i is the i^{th} attribute of the data point
- y_i is the i^{th} attribute of the test data point

The features used for pothole and speed bump detection were acceleration and gyroscope values in the x, y and, z directions resulting in a total of 6 features. For unsafe

Fig. 12 Weather data collection flowchart

driving behaviours detection additional features were used namely the mean, minimum, maximum, and standard deviation of the acceleration and gyroscope values. A total of 30 features were then obtained.

3.4.2 MLP Algorithm for Pothole, Speed Bump and Unsafe Driving Behaviours Detection

The most widely adopted type of neural network model in deep learning is the Multi-layered Perceptron (MLP). It was originally designed for image recognition but is now also used to solve complex problems including classification and regression. The multilayer perceptron is an artificial neural network that follows a feed-forward architecture, comprising three essential layers: the input layer, the hidden layer(s), and the output layer. The input layer is the starting layer of the network and takes in an input which is then used to produce

an output. The network has at least one hidden layer and its function is to perform all the computations and process the input data to produce meaningful results. The output layer displays the meaningful results [26].

The layers are interconnected and these connections are assigned weights which determine the importance of the connections. The weights are optimized through a process called backpropagation. Firstly, random values between -1 and 1 are given to the weights and the output is observed. The error which is the difference between the output and the expected output is propagated back through the network causing the weights to be readjusted. This process is repeated until the correct output is obtained. At this stage, the weights are the one that works correctly for the neural network.

In this paper, the gyroscope and accelerometer values were the inputs to the neural network. The number of input layers was six for pothole and speed bump detection and

Fig. 13 Sensor data collection flowchart

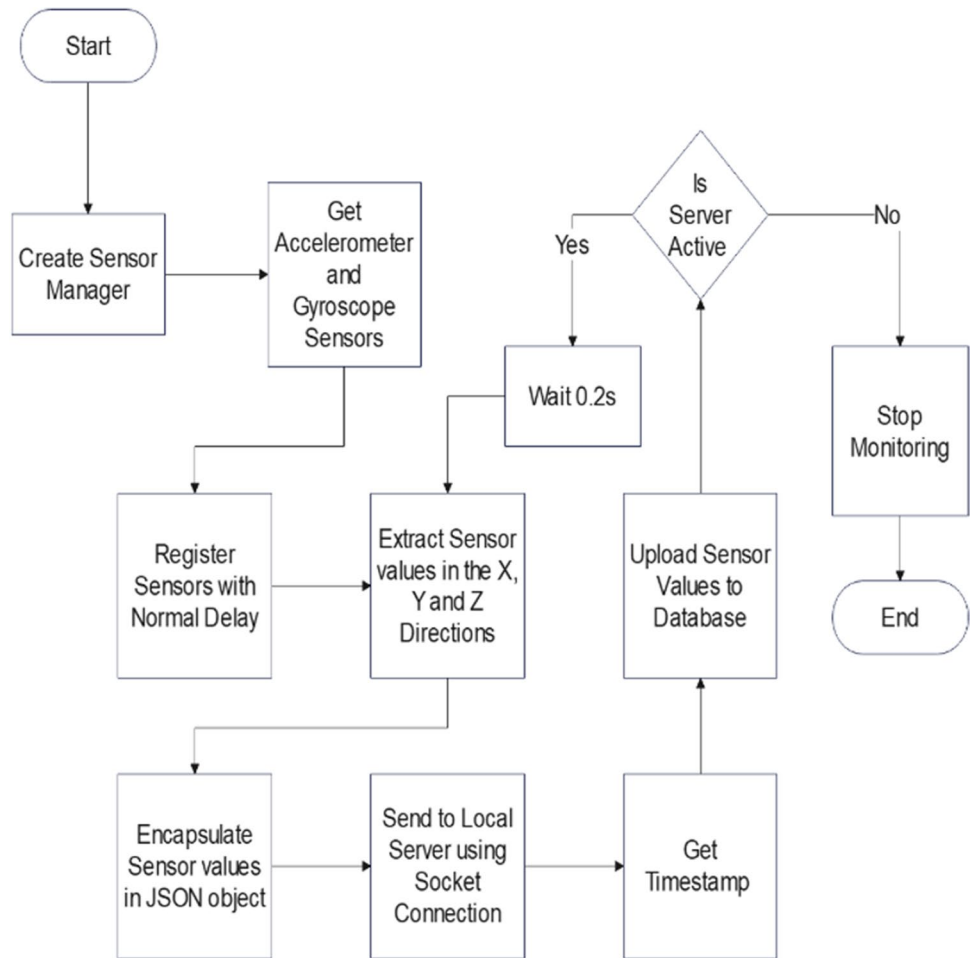


Table 3 Section of pothole dataset

accx	accy	...	gyroz	Class
0.0170	0.2039	...	0.007	0
0.0508	0.1940	...	0.009	0
0.0374	0.1912	...	0.009	0
0.0538	0.2777	...	0.002	0

Table 5 Section of speed bump dataset

accx	accy	...	gyroz	Class
-0.592	2.053	...	-0.057	0
1.108	1.921	...	-0.093	0
0.261	2.286	...	-0.083	0
0.492	2.088	...	-0.061	0

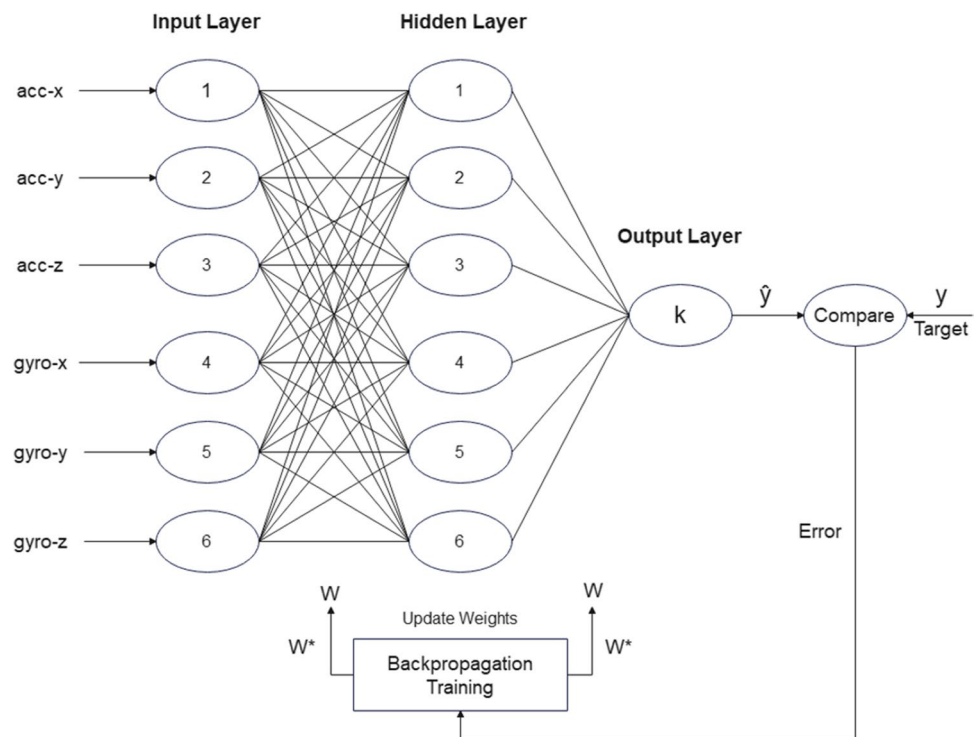
Table 4 Section of unsafe driving behaviours dataset

accx	accy	...	stdgyroz	Class
6.435	2.158	...	1.091	0
6.389	1.359	...	1.248	0
6.294	3.132	...	1.231	0
6.160	0.837	...	0.669	0

30 for unsafe driving behaviours. The number of hidden layers was obtained by trying different values and selecting the optimal one. Figure 14 shows an illustration of the backpropagation process in MLP.

In Fig. 14, the accelerometer and gyroscope values in the x, y, and z directions are set as inputs to the input layer. These values are denoted by acc-x, acc-y, acc-z, gyro-x, gyro-y, and gyro-z. Between the input layer and the hidden layer are the weights denoted by W. During the training of the model, an estimate of the output is obtained denoted by \hat{y} . It is compared with the expected output denoted by y

Fig. 14 Illustration of backpropagation process in MLP



to calculate the error. The error is propagated back to the network to obtain the updated weights denoted by W^* .

3.4.3 Algorithms for Estimating Braking Distance and Recommended Speed Using Predicted Rainfall

To estimate braking distance and recommended speed, the following steps were carried out:

- Predict rainfall intensity using MLR.
- Estimate the skid coefficient from the predicted rainfall intensity and speed of the vehicle using the Lagrange interpolation formula.
- Calculate braking distance using the braking distance formula which is based on the contact of the vehicle’s tyre with the pavement.
- Calculate recommended speed using a derived formula based on the road speed limit and skid coefficient.

The above procedures are explained in the following sections.

MLR Algorithm for Rainfall Forecasting Multiple linear regression is a type of predictive analysis used to find the relationship between a continuous dependent variable and several independent variables. It assumes the variables are linearly related and the independent variables have low correlation. In this paper, MLR was used to predict rainfall using a window size of 15 min [15]. The dependent variable is rainfall intensity and the independent variables are rainfall

intensity and cloudiness. The expression for predicting rainfall is modelled by Eq. (2)

$$R_{t+1} = \beta_0 + \beta_1 R_t + \beta_2 C_t + \varepsilon \tag{2}$$

where,

- R_{t+1} is the predicted rainfall at time $t + 1$
- $\beta_0, \beta_1, \beta_2$ are the regression coefficients
- R_t is the rainfall at time t
- C_t is the cloudiness at time t
- ε is the standard error

The rainfall prediction was utilized to estimate braking distance and recommended speed.

R_t which is the previous rainfall value at time t has the highest incidence in determining the rainfall at time R_{t+1} . Moreover, the cloudiness, C_t , as per the work carried out in [15], has shown to be another highly correlated parameter in determining the rainfall at time $t + 1$.

Lagrange Interpolation Formula The Lagrange interpolation formula is a method used to determine a polynomial that accurately passes through a given set of data points. The function obtained is an n^{th} -degree polynomial approximation to $f(x)$. It is useful for estimating new data points that falls within the range of a given group of data points [27]. The Lagrange interpolation formula for the n^{th} -degree polynomial is given in Eq. (3).

Table 6 Data points of skid coefficient at different vehicle speeds and rainfall intensities

Speed (km/hr)	20	40	60	80	100	120
R.I (mm/hr)						
5	0.6608	0.6287	0.5868	0.5346	0.4725	0.4007
10	0.6396	0.6075	0.5656	0.5133	0.4513	0.3789
15	0.6222	0.5901	0.5482	0.4959	0.4339	0.3620
20	0.6059	0.5737	0.5318	0.4801	0.4176	0.3457
25	0.5917	0.5596	0.5177	0.4654	0.4034	0.3310
30	0.5814	0.5493	0.5073	0.4556	0.3931	0.3212

$$f(x) = \frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_n)}f_0 + \frac{(x-x_0)(x-x_2)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_n)}f_1 + \dots + \frac{(x-x_0)(x-x_1)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\dots(x_n-x_{n-1})}f_n \tag{3}$$

To estimate the skid coefficient, the Lagrange interpolation formula was applied twice times on a set of data points containing values of skid coefficients at various rainfall intensities and vehicle speeds. The values of rainfall intensities were scaled down to better represent rainfall measurements in Mauritius. Table 6 shows the values of the skid coefficient at different rainfall intensities and vehicle speeds which were obtained from [17].

The Lagrange formula was first applied to obtain the polynomial expressions which approximate the data points of the skid coefficient against the speed of each value of rainfall intensity as shown in Table 6. The approximated expressions are shown as follows:

$$\mu(v)_1 = 0.401v^5 + 0.473v^4 + 0.535v^3 + 0.587v^2 + 0.629v + 0.661 \tag{4}$$

$$\mu(v)_2 = 0.379v^5 + 0.451v^4 + 0.513v^3 + 0.566v^2 + 0.607v + 0.640 \tag{5}$$

$$\mu(v)_3 = 0.362v^5 + 0.434v^4 + 0.496v^3 + 0.548v^2 + 0.590v + 0.622 \tag{6}$$

$$\mu(v)_4 = 0.346v^5 + 0.412v^4 + 0.480v^3 + 0.532v^2 + 0.574v + 0.606 \tag{7}$$

$$\mu(v)_5 = 0.331v^5 + 0.403v^4 + 0.465v^3 + 0.518v^2 + 0.560v + 0.592 \tag{8}$$

$$\mu(v)_6 = 0.321v^5 + 0.393v^4 + 0.456v^3 + 0.507v^2 + 0.549v + 0.581 \tag{9}$$

where,

- $\mu(v)_1, \mu(v)_2, \dots, \mu(v)_6$ are the expressions for skid coefficient at 5 mm/hr, 10 mm/hr, ..., 30 mm/hr rainfall intensities.
- v is the speed of the vehicle

To find the skid coefficient at a particular speed and rainfall intensity, the value of speed was replaced in the approximated expressions to obtain another set of data points. The data points obtained are shown in Table 7.

The Lagrange formula was then applied a second time on the set of data points to obtain a polynomial approximation of the skid coefficient against rainfall intensity at that particular speed. The approximated expression is shown in Eq. (10).

$$\mu(r, v) = \frac{(r-10)(r-15)(r-20)(r-25)(r-30)}{(r-10)(5-15)(5-20)(5-25)(5-30)}\mu(v)_1 + \frac{(r-5)(r-15)(r-20)(r-25)(r-30)}{(10-5)(10-15)(10-20)(10-25)(10-30)}\mu(v)_2 + \frac{(r-5)(r-10)(r-20)(r-25)(r-30)}{(15-5)(15-10)(15-20)(15-25)(15-30)}\mu(v)_3 + \frac{(r-5)(r-10)(r-15)(r-25)(r-30)}{(20-5)(20-10)(20-15)(20-25)(20-30)}\mu(v)_4 + \frac{(r-5)(r-10)(r-15)(r-20)(r-30)}{(25-5)(25-10)(25-15)(25-20)(20-30)}\mu(v)_5 + \frac{(r-5)(r-10)(r-15)(r-20)(r-25)}{(30-5)(30-10)(30-15)(30-20)(30-25)}\mu(v)_6 \tag{10}$$

The value of rainfall intensity is replaced in the approximated expression to obtain the value of skid coefficient.

Skid Resistance Skid resistance is the force induced when a tyre is stopped from rotating and instead slides along the pavement surface. Skid resistance is dependent on several factors such speed of the wheel, pavement wetness, temperature, tire wear, etc. Figure 15 illustrates the concept of

Table 7 Data points of skid coefficient at different rainfall intensities

R.I (mm/hr)	5	10	15	20	25	30
Skid Coefficient	$\mu(v)_1$	$\mu(v)_2$	$\mu(v)_3$	$\mu(v)_4$	$\mu(v)_5$	$\mu(v)_6$

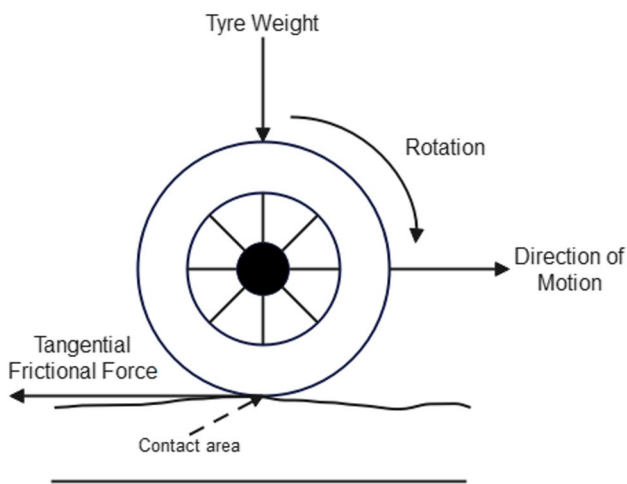


Fig. 15 Illustration of skid resistance

skid resistance at the contact area between the tyre and the pavement.

Skid resistance is of prime importance for road agencies and institutions as it has a direct effect on the number of road accidents especially during wet weather conditions. During such conditions, the presence of water on road surfaces reduces the grip between the tires and the pavement which results in less contact area at the tyre-pavement interface. Consequently, skid resistance decreases and the risk of road accidents on wet pavements is greater as compared to dry pavements [17]. In this paper, skid coefficient is used to determine the braking distance of a vehicle under wet conditions. Skid coefficient is a measure of the skid resistance and is a number between 0 and 1.

When a vehicle is moving at a specific speed and the brakes are suddenly applied, the vehicle will eventually come at rest after travelling a certain distance. This distance is called the braking distance. It is dependent on several factor such as the vehicle speed, the driver’s reaction time and the road surface condition. Equation (11) shows the formula used for calculating the braking distance.

$$d = v\tau + \frac{v^2}{2g\mu} \tag{11}$$

where,

- d is the braking distance in metres
- v is the speed of the vehicle in Kmphr.^{-1}
- τ is average reaction time of a driver in seconds
- g is the acceleration due to gravity in ms.^{-2}
- μ is the skid coefficient calculated using Eq. (10)

An expression for calculating a safe recommended speed is derived as follows:

1. The maximum braking distance that a vehicle can have on a particular road is calculated. It is assumed that the maximum braking distance occurs when the speed of the vehicle is equal to the road speed limit and there is no rainfall. The following expression is obtained when replacing in Eq. (11):

$$d_{max} = v_{max}\tau + \frac{v_{max}^2}{2g\mu_0} \tag{12}$$

where, μ_0 is obtained by replacing $r = 0$ and $v = v_{max}$ in Eq. (10) and v_{max} is the road speed limit.

2. To get the recommended speed during rainfall, d_{max} and the value of skid coefficient during rainfall is replaced in Eq. (10). The following expression is obtained:

$$d_{max} = v_{rec}\tau + \frac{v_{rec}^2}{2g\mu_r} \tag{13}$$

where, μ_r is obtained by replacing $r = predictedrain$ and $v = v_{max}$ in Eq. (10) and v_{rec} is the recommended speed.

3. Solving for v_{rec} using the quadratic formula, the following expression for the recommended speed is obtained:

$$v_{rec} = \frac{-2\tau g\mu_r + \sqrt{(2\tau g\mu_r)^2 + 8d_{max}}}{2} \tag{14}$$

3.4.4 Calculation of Road and Driver Ratings

The driver rating provides an objective assessment of the person’s driving performance during their journey and is based on the fraction of time the person exceeds the recommended speed. On the other hand, the road rating indicates the condition and quality of the road itself and takes into account the number of road events such as potholes and speed bumps which were encountered during the driver’s journey. Equations (15) and (16) shows the formulae used to calculate the driver’s speed deviation and number of road events during a driver’s journey respectively. The basis for allocating the driver rating is that the driver is rewarded for keeping a speed below the recommended speed and is penalised for going above the recommended speed. For the road rating, it is allocated based on the number of potholes and speed bumps present on the road indicating that the road is poorly maintained if there are too many potholes and speed bumps.

$$S_D = \frac{1}{N} \sum_{s=1}^N (v_s - v_{rec}) \times 100\% \tag{15}$$

Table 8 Driver rating allocation

Speed deviation (%)	Driver rating
0–4	5
4–8	4
8–12	3
12–16	2
16+	1

Table 9 Road rating allocation

N° of road events/ Km	Road rating
0–2	5
3–5	4
6–8	3
9–11	2
12+	1

$$E = \frac{N^{\circ} \text{ of Potholes} + N^{\circ} \text{ of Speed bumps}}{d} \tag{16}$$

where,

- S_D is the speed deviation in percentage
- v_s is the actual speed at sampling instance s
- v_{rec} is the recommended speed
- N is the number of sampling instances
- E is the N° of road events per Km
- d is the distance travelled in Km

Table 8 shows how the driver rating is allocated based on the speed deviation.

Table 9 shows how the road rating is allocated based on the N° of road events per Km.

4 Performance of Machine Learning Algorithms

In this section, the performance of the MLR algorithm used in this application for rainfall prediction will be analysed. Performance analysis of KNN and MLP algorithms will be carried out to evaluate their effectiveness in detecting potholes, unsafe driving behaviours and speed bumps. The analysis consists of a cross-validation test where the dataset used to train the models is evaluated and a real-time test where test drives were carried out. Finally, the algorithms used for estimating braking distance and recommended speed will be analysed.

Some of the performance parameters used to evaluate the algorithms are expressed as follows:

$$MAPE = \frac{1}{N} \sum_{j=1}^N \frac{|A_j - P_j|}{|A_j|} \times 100\% \tag{17}$$

where,

- $MAPE$ is the Mean Absolute Percentage Error
- MSE is the Mean Square Error
- $RMSE$ is the Root Mean Square Error
- MAE is the Mean Absolute Error
- A_j is the actual value
- P_j is the predicted value
- N is the number of data samples

4.1 Performance of MLR for Rainfall Prediction

To evaluate the performance of the MLR algorithm for rainfall prediction, data was collected for a period of eight hours for five different locations under moderate rainfall conditions. Rainfall is predicted using the weather parameters given in Table 10.

The prediction was carried out for the next 15 min by inputting the collected data in Eq. (2). Table 11 summarizes the data collected for the locations considered.

Table 12 shows the experimental results when evaluating the MLR algorithm for rainfall prediction.

The MLR algorithm was chosen as it is an algorithm which is computationally efficient compared to other algorithms such as polynomial regression and neural networks which may not necessarily provide better results. The MAPE will be used to interpret the results obtained in Table 12. MAPE gives a more intuitive understanding of the performance of the forecasting model.

Table 10 Weather parameters used for rainfall prediction

Dependent variable	Independent variables
Rainfall	<ul style="list-style-type: none"> • Rainfall • Cloudiness

Table 11 Summary of weather data collected

Observation Period	8 h
N° of Samples	931
Weather parameters	<ol style="list-style-type: none"> 1. Rainfall – (mmhr⁻¹) 2. Cloudiness – (%)
Location Names	<ul style="list-style-type: none"> • Goodlands • Pamplermousses • Port Louis • Grand Bay • Piton

Table 12 Experimental results for evaluating MLR for rainfall prediction

Location	RMSE	MSE	MAE	MAPE
Goodlands	2.931	8.592	1.108	13.38%
Pamplemousses	3.167	10.03	1.455	15.56%
Port Louis	2.532	6.409	0.899	9.002%
Grand Bay	3.544	12.56	1.825	16.89%
Piton	1.622	2.631	0.457	5.672%
Average	2.836	8.044	1.149	12.10%

From the results obtained in Table 12, it can be seen that the errors obtained for rainfall prediction are significant but not too high ranging from 5.672% to 16.89% with an average of 12.10% for the five locations. The high error can be attributed to the fact that rainfall patterns can vary over a short time frame and can be localized to affect only certain areas.

It can be deduced that MLR is an algorithm which performs reasonably well in predicting rainfall when referred to previous works. In [28], it is observed that MLR provides better accuracy than other existing algorithms. However, in [29], where the capability of linear and nonlinear regression techniques was analysed, it was found that MLR gave poorer results as rainfall has a nonlinear dependence on variables such as temperature and humidity. Although the errors were high, this doesn't indicate that the MLR algorithm couldn't accurately model the data. This is because rainfall is a parameter which is challenging to predict in nature.

Table 13 Hyper parameters of KNN and MLP algorithms

Algorithm	Scenario	Parameters
KNN	Pothole detection	<ul style="list-style-type: none"> • N° of Neighbours = 6 • Distance Weighting = Euclidean
	Speed bump detection	<ul style="list-style-type: none"> • N° of Neighbours = 4 • Distance Weighting = Euclidean
	Unsafe driving behaviours detection	<ul style="list-style-type: none"> • N° of Neighbours = 6 • Distance Weighting = Euclidean
MLP	Pothole detection	<ul style="list-style-type: none"> • N° of Input nodes = 6 • N° of hidden nodes = 12 • Activation function = ReLU • Learning rate = 0.1 • Max epochs = 100
	Speed bump detection	<ul style="list-style-type: none"> • N° of Input nodes = 6 • N° of hidden nodes = 6 • Activation function = Sigmoid • Learning rate = 0.1 • Max epochs = 100
	Unsafe driving behaviours detection	<ul style="list-style-type: none"> • N° of Input nodes = 30 • N° of hidden nodes = 30 • Activation function = Sigmoid • Learning rate = 0.1 • Max epochs = 100

4.2 Performance of KNN and MLP for Pothole, Speed Bump and Unsafe Driving Behaviours Detection

To evaluate the performance of KNN and MLP algorithms for pothole, speed bump and unsafe driving behaviours detection, two different methods were used. In the first method, the datasets used to train the models were tested using a cross-validation technique. In the second method, real-time drive tests were carried out to determine the accuracy of the algorithms. The datasets for potholes and unsafe driving behaviours were both obtained from the Kaggle website [22, 23]. The speed bumps dataset was generated by collecting data using a smartphone during drive tests conducted on various speed bumps.

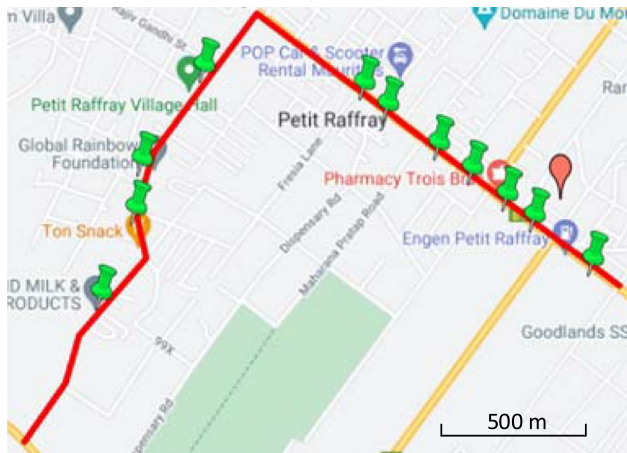
The hyper parameters used to train the KNN and MLP algorithms were obtained by testing different combinations of values. The optimum values were then selected based on the highest accuracy obtained. Table 13 summarized the hyper parameters for the KNN and MLP algorithms (Table 13).

Table 14 Cross-validation results for KNN algorithm

Dataset	N° of Instances	Accuracy
Potholes	9859	95.34%
Speed Bumps	660	73.48%
Unsafe Driving Behaviours	2301	98.74%

Table 15 Cross-validation results for MLP algorithm

Dataset	N° of Instances	Accuracy
Potholes	9859	95.38%
Speed Bumps	660	72.27%
Unsafe Driving Behaviours	2301	89.44%

**Fig. 16** Route for verifying speed bump and unsafe driving behaviours detection algorithms

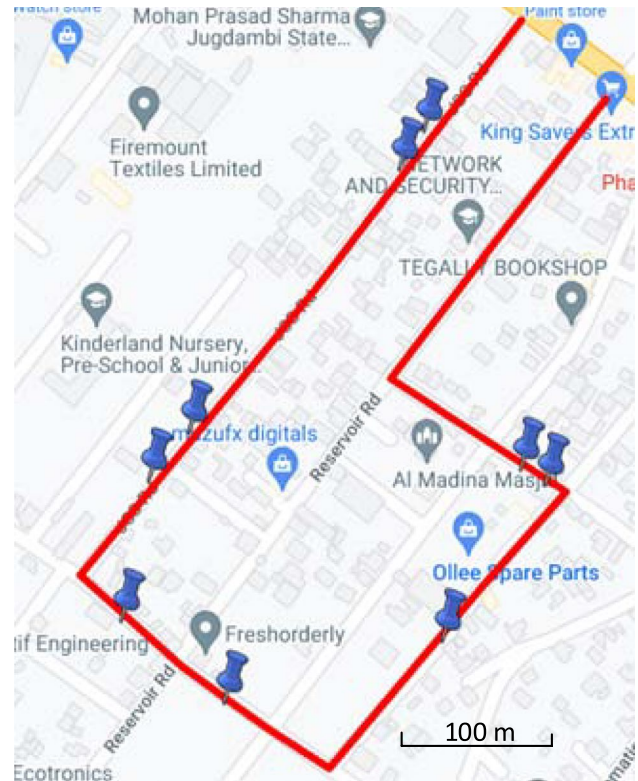
4.2.1 Cross-Validation of Datasets

When performing the cross-validation, the datasets are divided into the training sets and the testing sets. The ratio of training sets to testing sets is set to 7:3. The training sets are used to build the classification models. The testing sets are then used to find the accuracy of the models. Tables 14 and 15 show the cross-validation results for KNN and MLP algorithms respectively.

From Tables 14 and 15, the accuracy for pothole detection is high for both KNN and MLP algorithms. This can be attributed to the large dataset used for training the models. For speed bump detection, an average accuracy is observed and this is explained by the small dataset which was collected to train the models. The accuracy for detecting unsafe driving behaviours is high as the number of attributes used to train the models is large (30). However, MLP performed poorly compared to KNN. MLP is an algorithm which gives better results when the dataset is very large.

4.2.2 Real-Time Drive Tests

Drive tests are carried out to determine if the algorithms will be effective under real-world conditions. They help determine whether the algorithms can be generalized to different road surface conditions and vehicles instead of overfitting the particular datasets used to train the model.

**Fig. 17** Route for verifying pothole detection algorithms**Table 16** Drive test results for KNN algorithms

Parameter	N° of Occurrences	Accuracy
Potholes	21	80.9%
Speed Bumps	25	72.0%
Unsafe Driving Behaviours	34	88.2%

Table 17 Drive test results for MLP algorithms

Parameter	N° of Occurrences	Accuracy
Potholes	21	80.9%
Speed Bumps	25	68.0%
Unsafe Driving Behaviours	34	82.3%

The drive tests were carried out on the roads of Mauritius in the region of Petit Raffray and Goodlands. A car was used to perform the drive tests. The speed limit for the routes was 60 km/hr while the car was driven at around 40 km/hr. The drive tests were carried out at two o'clock in the afternoon when there was no traffic congestion. Figures 16 and 17 show the routes selected to carry out the drive tests, one for the speed bump and unsafe driving behaviours detection algorithms and the other for the pothole detection algorithms. A green marker in Fig. 16

indicates the location of a speed bump and a blue marker in Fig. 17 indicates the location of a pothole.

After completing the drive tests, the following results were obtained for KNN and MLP algorithms as shown in Tables 16 and 17.

From the results shown in Tables 16 and 17, it can be seen that both KNN and MLP algorithms performed well. The accuracy was lower for the drive tests but not significantly. For pothole detection, both algorithms have the same accuracy. However, for speed bump and unsafe driving behaviours detection, KNN gave better results.

Both the cross-validation tests and the drive tests showed that KNN and MLP algorithms can accurately model the data. The accuracy for speed bump detection is low with both tests. One reason to explain this is the introduction of human errors such as timing errors when collecting data for the dataset. When comparing with similar works, it can be argued that MLP and KNN are algorithms which perform well for road condition monitoring.

4.3 Performance of Braking Distance and Recommended Speed Estimation Algorithm

To obtain the performance of the algorithm for estimating braking distance and recommended speed, the values of braking distance and recommended speed were predicted after a period of 15 min at different speed limits using the data collected for rainfall. Equation (11) was used to obtain the predicted braking distance and Eq. (14) was used to obtain the predicted recommended speed.

Figure 18 shows the variation of braking distance with rainfall intensity at different vehicle speeds.

Figure 19 shows the variation of recommended speed with rainfall intensity at different speed limits.

The predicted values of braking distance and recommended speed were then compared with the actual values

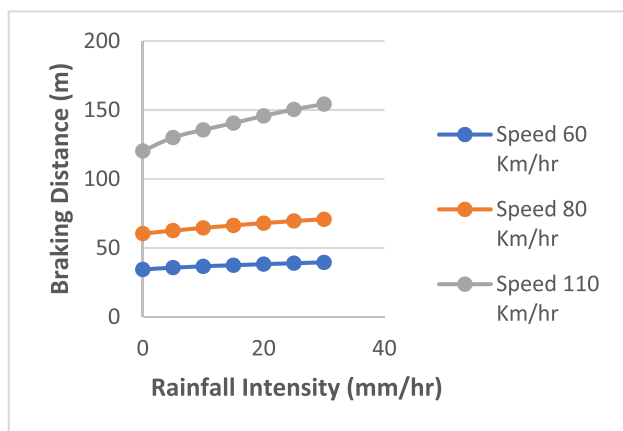


Fig. 18 Variable of braking distance with rainfall intensity

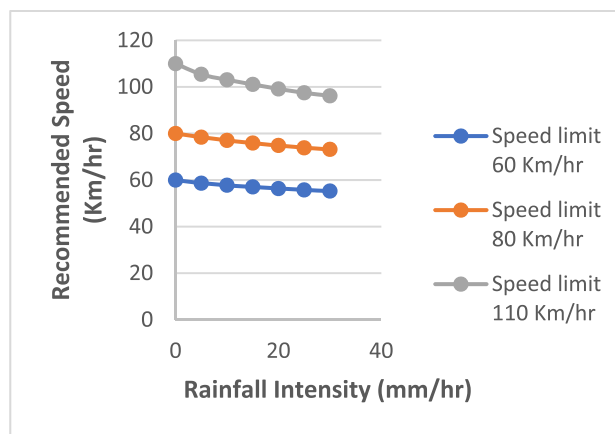


Fig. 19 Variation of recommended speed with rainfall intensity

to obtain the performance metrics. Since the algorithm relies on the predicted values of rainfall, the performance obtained is directly related to the accuracy of MLR used for rainfall prediction. Table 18 shows the results obtained for the algorithm.

From the results obtained in Table 14, it is observed that the error obtained for recommended speed ranges from 0.591% to 0.987% which is very low. The explanation for this is that recommended speed has low variation with respect to rainfall. In the case of braking distance, the error is relatively high ranging from 11.55% to 18.13% and this is attributed to braking distance being proportional to the square of speed that is a small change in speed will result in a large change in braking distance. A trend is also observed where a low error occurs at a low-speed limit and a high error occurs at a high-speed limit.

4.4 Interpretation of Driver and Road ratings

By considering the drive tests which were carried out to verify the algorithms for the pothole and speed bump detection,

Table 18 Results of braking distance and recommended speed estimation algorithm

Parameter	RMSE	MSE	MAE	MAPE
Speed limit = 110 km/hr				
Recommended speed	2.573	6.621	1.019	0.987%
Braking distance	147.1	21,643	35.33	18.13%
Speed limit = 80 km/hr				
Recommended speed	1.323	1.750	0.483	0.628%
Braking distance	53.86	2900	16.27	14.38%
Seed limit = 60 km/hr				
Recommended speed	0.892	0.796	0.342	0.591%
Braking distance	26.47	700.5	10.57	11.55%

the speed deviation obtained was 0.83% using Eq. (15) and the number of events per km obtained was 10.6 using Eq. (16). This results in a driver rating of 5 and a road rating of 2.

5 Conclusion

In this paper, a system for monitoring road conditions and unsafe driving behaviours in real-time was implemented. Essentially the system consisted of local server and a mobile application. The local server is a GUI desktop application created using Java. The server was tasked to perform classifications for detecting potholes, speed bumps and unsafe driving behaviour using sensor data collected from the mobile application and then send the results to the mobile application. The algorithms used for classification were KNN and MLP. The results showed an average accuracy of 80.9% for pothole detection, 70% for speed bump detection and 85.3% for unsafe driving behaviours detection. In the work carried out in [30], an accuracy of 85% for pothole detection was obtained using deep learning method. The results obtained in [31] showed an accuracy of 78.39% using KNN and 76.69% using MLP for driving behaviour recognition. The low accuracy observed for speed bump detection is explained by the use of a relatively smaller dataset due to relatively shorter roads found in Mauritius. Moreover, some errors due to reaction time may have been introduced when collecting the sensor data for the dataset. The reaction time here refers to τ in Eq. (11) and the error due to it can be reduced by repeating the experiment more times or using a larger data set. The braking distance and recommended speed during rainy weather were estimated. This was done by first predicting rainfall intensity using MLR. The braking distance was then calculated based on the rainfall intensity and speed of the vehicle. As for the recommended speed, a formula was derived based on the rainfall intensity and road speed limit. The MAPE for the above were as follows: 12.1% for rainfall prediction, 14.7% for braking distance, and 0.735% for recommended speed. Finally, a method for providing ratings for the driver's journey was devised. It was based on the road quality and the performance of the driver. Future works can be explored to improve the performance of the system and accuracy of the algorithms selected. Firstly, by making use of larger datasets, the accuracy of monitoring road conditions will significantly increase. Algorithms such as MPR and LSTM would provide better results for rainfall prediction as MPR takes into account the non-linear dependence on predictor variables while LSTM is more suitable for time-series data.

Abbreviations API: Application Programming Interface; CART: Classification And Regression Tree; CNN: Convolutional Neural Network;

CSV: Comma Separated Values; GBDT: Gradient-Boosted Decision Tree; GPS: Global Positioning System; GUI: Graphical User Interface; HTML: Hyper Text Markup Language; HTTP: Hyper Text Transfer Protocol; IDE: Integrated Development Environment; IP: Internet protocol; JSON: JavaScript Object Notation; KNN: K Nearest Neighbours; LSTM: Long Short-Term Memory Networks; MAPE: Mean Absolute Percentage Error; MLP: Multi-Layer Perceptron; MLR: Multiple Linear Regression; MPR: Multiple Polynomial Regression; SDK: Software Development Kit; SQL: Structured Query Language; SVM: Support Vector Machine; UI: User Interface; URL: Uniform Resource Locator

Acknowledgements The authors are thankful to the University of Mauritius for providing the necessary facilities for conducting this research.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

References

- Govmu.org: (2022). Available: https://statsmauritius.govmu.org/Pages/Statistics/ESI/Transport/RT_RTA_Jan-Jun22.aspx. Accessed 7 July 2023
- WHO, World Health Organization: (2022). Available: <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>. Accessed 13 July 2023
- Wes, M.: WorldBank.org. (2015). Available: <https://www.worldbank.org/en/news/opinion/2015/01/05/prevent-road-accidents-the-swedish-example>. Accessed 13 July 2023
- R. a. I. Magazine: (2018). Available: <https://roadsonline.com.au/saving-lives-on-country-roads-the-potential-for-flexible-safety-barriers-on-au-roads/>. Accessed 13 July 2023
- Ahmed, K.R.: Smart pothole detection using deep learning based on dilated convolution. *Sensors* **21**(24), 8406 (2021). <https://doi.org/10.3390/s21248406>
- Aparna, Bhatia, Y., Rai, R., Gupta, V.: Convolutional neural networks based potholes detection using thermal imaging. *J. King Saud Univ.* **34**(3), 578–588 (2019). <https://doi.org/10.1016/j.jksuci.2019.02.004>
- Wu, C., Wang, Z., Hu, S., Lepine, J., Na, X., Ainalis, D., Stettler, M.: An Automated machine-learning approach for road pothole detection using smartphone sensor data. *Sensors* **20**(19), 5564 (2020). <https://doi.org/10.3390/s20195564>
- Pawar, K., Jagtap, S., Bhoir, S.V.: Efficient pothole detection using smartphone sensors. *ITM Web Conf.* **32**, 03013 (2020). <https://doi.org/10.1051/itmconf/20203203013>
- Padilla, C., Tejada, J.G., Monteagudo, C.L., Gonzales, F.A., Baez, O.M., Torteya, M., Tejada, A.G., Olague, A., Garcia, L., Rosales, G.: Speed bump detection using accelerometric features: a genetic algorithm approach. *Sensors* **18**(2), 443 (2018). <https://doi.org/10.3390/s18020443>
- Casati, J. B. C., Altafim, R. A. C., Altafim, A. P.: Vibration detection of vehicle impact using smartphone accelerometer data and long-short term memory neural network. **2**(1), (2020). <https://doi.org/10.48011/asba.v2i1.1274>
- Lattanzi, E., Freshchi, V.: Machine learning techniques to identify unsafe driving behavior by means of in-vehicle sensor data. *Expert Syst. Appl.* **176**, 114818 (2021). <https://doi.org/10.1016/j.eswa.2021.114818>
- Yi, N., Zhenming, L., Yunxiao, F.: Analysis of truck drivers' unsafe driving behaviors using four machine learning methods.

- Int. J. Ind. Ergon. **86**, 103192 (2021). <https://doi.org/10.1016/j.ergon.2021.103192>
13. Liyew, C., Melese, H. A.: Machine learning techniques to predict daily rainfall amount. *J. Big. Data.* **8**(1), (2021). <https://doi.org/10.1186/s40537-021-00545-4>
 14. Endalie, D., Haile, G., Taye, W.: Deep learning model for daily rainfall prediction: case study of Jimma, Ethiopia. *Water Supply* **22**(3), 3448–3461 (2021). <https://doi.org/10.2166/ws.2021.391>
 15. Fowdur, T.P., Rosun, M.N.-U.-D.I.N.: A real-time collaborative machine learning based weather forecasting system with multiple predictor locations. *Array* **14**, 100153 (2022). <https://doi.org/10.1016/j.array.2022.100153>
 16. Ali, A.K., Omid, R., Amir, S.A.N., Sid, M.B.: Effect of adverse weather conditions on vehicle braking distance of highways. *Civ. Eng. J.* **4**(1), 46–57 (2018). <https://doi.org/10.28991/cej-030967>
 17. Tianchi, T., Kumar, A., Cor, K., Athanasios, S., Sandra, E.: A finite element study of rain intensity on skid resistance for permeable asphalt concrete mixes. *Constr. Build. Mater.* **220**, 464–475 (2019). <https://doi.org/10.1016/j.conbuildmat.2019.05.185>
 18. Oche, A.E., Gareth, E., Mark, G.G., Gregory, I.: Real-time machine learning-based approach for pothole detection. *Exp. Syst. Appl.* **184**, 115562 (2021). <https://doi.org/10.1016/j.eswa.2021.115562>
 19. Marques, J., Alves, R., Oliveira, R., Mendonça, M., Souza, J. R.: An evaluation of machine learning methods for speed-bump detection on a GoPro dataset. *An. Acad. Bras. Ciênc.* **93**(1), (2021). <https://doi.org/10.1590/0001-3765202120190734>
 20. Michal, M., Christopher, C.Y.: Detecting aggressive driving patterns in drivers using vehicle sensor data. *Transp. Res. Interdiscip. Perspect.* **14**, 100625 (2022). <https://doi.org/10.1016/j.trip.2022.100625>
 21. Ahmed, A., Karim, E.-B., Tae, J. K.: Effects of inclement weather events on road surface conditions and traffic safety an event-based empirical analysis framework. *Transp. Res. Rec J. Transp. Res. Board.* **2676**(10), 51–62 (2022). <https://doi.org/10.1177/03611981221088588>
 22. Tiwari and Shashwat: Kaggle. (2021). Available: <https://www.kaggle.com/datasets/shashwatwork/driving-behavior-dataset>. Accessed 29 June 2023
 23. Goes and Dexter: Kaggle. (2020). Available: <https://www.kaggle.com/datasets/dextergoes/pothole-sensor-data?resource=download>. Accessed 29 June 2023
 24. Michael, S., Tan, P.-N.: The top ten algorithms in data mining, pp. 151–159. Crc Press, Boca Raton (2009)
 25. IBM: Available: <https://www.ibm.com/topics/knn>. Accessed 13 June 2023
 26. Hassan, R., Mohammed, A., Janati, I., Youssef, G., Mohamed, E.: Multilayer perceptron: architecture optimization and training. *Int. J. Interact. Multimedia. Artif. Intell.* **41**(1), 26 (2016). <https://doi.org/10.9781/ijimai.2016.415>
 27. Steven C.C., Raymond, P.C.: Numerical methods for engineers, 6th edn. McGraw-Hill Science/Engineering/Math (2009)
 28. Sirisha, D.: Predicting rainfall using machine learning techniques. *IJIRT* **8**(4), (2021). <https://doi.org/10.36227/techrxiv.14398304.v1>
 29. Ilaboya, I. R. Igbinedion, O. E.: Performance of multiple linear regression (MLR) and artificial neural network (ANN) for the prediction of monthly maximum rainfall in Benin City, Nigeria. *Int. J. Eng. Sci. Appl.* **3**, (2019). <https://dergipark.org.tr/tr/download/article-file/682360>. Accessed June 2023
 30. Anas, A.-S., Rami, A., Samir, B.: Real-time pothole detection using deep. (2021). <https://doi.org/10.48550/arXiv.2107.06356>
 31. Wen-Hui, C., Yu-Chen, L., Wei-Hao, C.: Comparisons of machine learning algorithms for driving behavior recognition using in-vehicle CAN bus data. *IEEE Access* (2019). <https://doi.org/10.1109/ICIEV.2019.8858531>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Dr. T.P. Fowdur received his BEng (Hons) degree in Electronic and Communication Engineering with first class honours from the University of Mauritius in 2004. He was also the recipient of a Gold medal for having produced the best degree project at the Faculty of Engineering in 2004. In 2005 he obtained a full-time PhD scholarship from the Tertiary Education Commission of Mauritius and was awarded his PhD degree in Electrical and Electronic Engineering in 2010 by the University of Mauritius. He is also a Registered Chartered Engineer of the Engineering Council of the UK, Fellow of the Institute of Telecommunications Professionals of the UK and Senior Member of the IEEE. He joined the University of Mauritius as an academic in June 2009 and is presently an Associate Professor at the Department of Electrical and Electronic Engineering of the University of Mauritius. His research interests include Mobile and Wireless Communications, Multimedia Communications, Networking and Security, Telecommunications Applications Development, Internet of Things and AI. He has published several papers in these areas and is actively involved in research supervision, reviewing of papers and also organizing international conferences.

Mr Mohammed Fayeze Hawseer holds a Degree in Telecommunications Engineering with Networking with first class honours from the University of Mauritius. He is presently working as an Engineer at Emtel Mauritius Limited. His research interests include Machine learning, Real-time Systems and Mobile Communications.